

## 實驗 3- 認識 LoRa 通訊

### ✦學習目標：

1. 了解 LoRa 模組傳送資料的方式。

### ✦實驗概要

本次實驗使用到的模組為 433MHz 的 LoRa 模組，主要用到 RFM95 模塊(圖 1 的標示 4)，使用 SPI 介面，如圖 2，但本次用到的模組是用 UART 與 ARDUINO 連接，因此程式會先經過一個 SPI 轉 UART 的動作。

而 RFM95 不像之前使用的 WiFi 及藍芽模組一樣有 AT 指令可以設定及查看模組狀態(市面上有些 LoRa 模組是有 AT 指令的)，因此若是要指定傳送與接收位址便需要較複雜的程式技巧來完成，而本次僅實作簡單的資料傳送，並無指定傳送位址，因此若是有多個 LoRa 模組以在此次程式下運行會分不清究竟是哪一個模組傳送/接收了訊息。

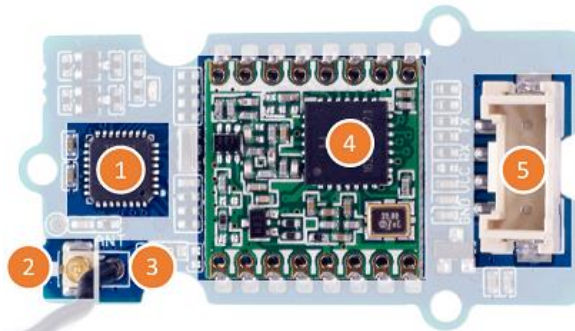


圖 1. LoRa 模組區塊圖(1. ATmega168 MCU 2. MHF Connector 3. Wire Antenna 4. RFM95 Module 5. Grove Interface)。



圖 2. LoRa 模組背面。

### ✦實驗 1- 從端程式步驟

1. 將從端程式燒錄 Arduino。

(1) 如下圖連接 LoRa 模組(VCC、GND、TX(2)、RX(3))。

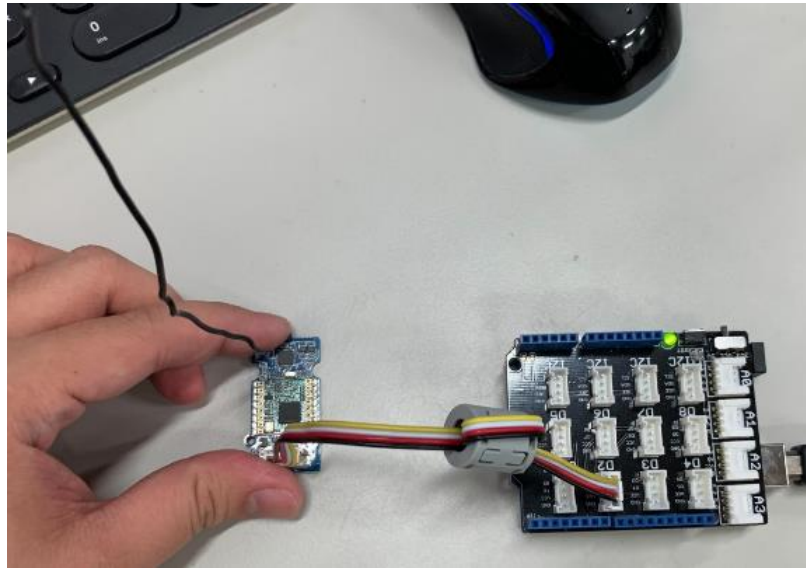


圖 3. Lora 模組接線圖。

(2) 打上以下程式碼。

//////////從端程式//////// 負責傳遞訊息

```
#include <RH_RF95.h>
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial SSerial(2, 3); // RX, TX
```

```
#define COMSerial SSerial
```

```
#define ShowSerial Serial
```

```
RH_RF95<SoftwareSerial> rf95(COMSerial);
```

```
void setup() {
```

```
    ShowSerial.begin(115200);
```

```
    ShowSerial.println("RF95 client test.");
```

```
    if (!rf95.init()) {
```

```
        ShowSerial.println("init failed");
```

```
        while (1);
```

```
    }
```

```
    rf95.setFrequency(433.0);
```

```
}
```

```
void loop() {
```

```

ShowSerial.println("Sending to rf95_server");
// Send a message to rf95_server

/*if (ShowSerial.available()) {
    val = ShowSerial.read();
}
ShowSerial.print(val);*/

uint8_t data[] = "hello"; //編輯傳送訊息
rf95.send(data, sizeof(data));

rf95.waitPacketSent();

// Now wait for a reply
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);
//回復訊息
/*
if (rf95.waitForAvailable(3000)) {
    // Should be a reply message for us now
    if (rf95.recv(buf, &len)) {
        ShowSerial.print("got reply: ");
        ShowSerial.println((char*)buf);
    } else {
        ShowSerial.println("recv failed");
    }
} else {
    ShowSerial.println("No reply, is rf95_server running?");
}
*/
delay(1000);
}

```

(3) 編譯成功後即可從序列埠看到運行狀態，如圖 4 所示。

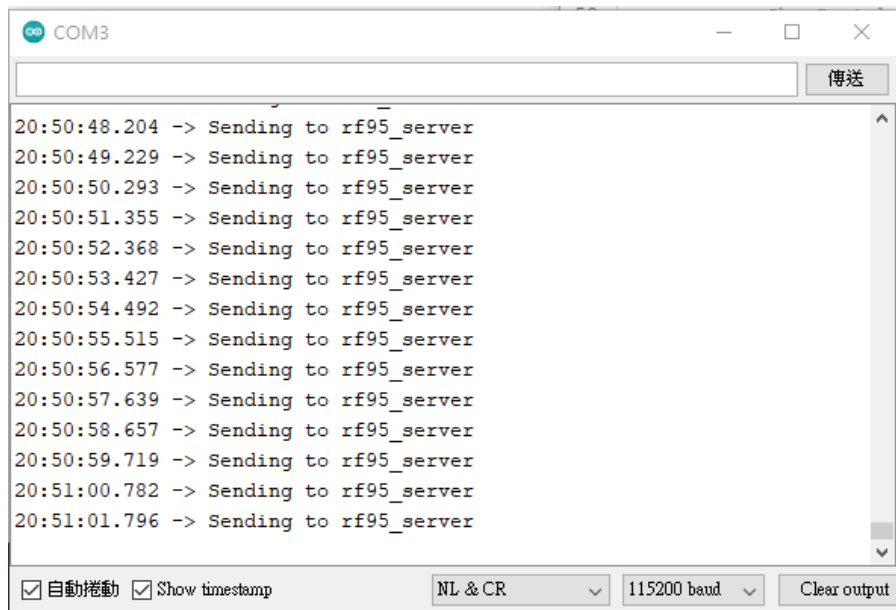


圖 4.從端發送訊息

## †實驗 2- 從主端程式步驟

1. 將從端程式燒錄至 Arduino。

(1) 如下圖連接 LoRa 模組(VCC、GND、TX(2)、RX(3))。

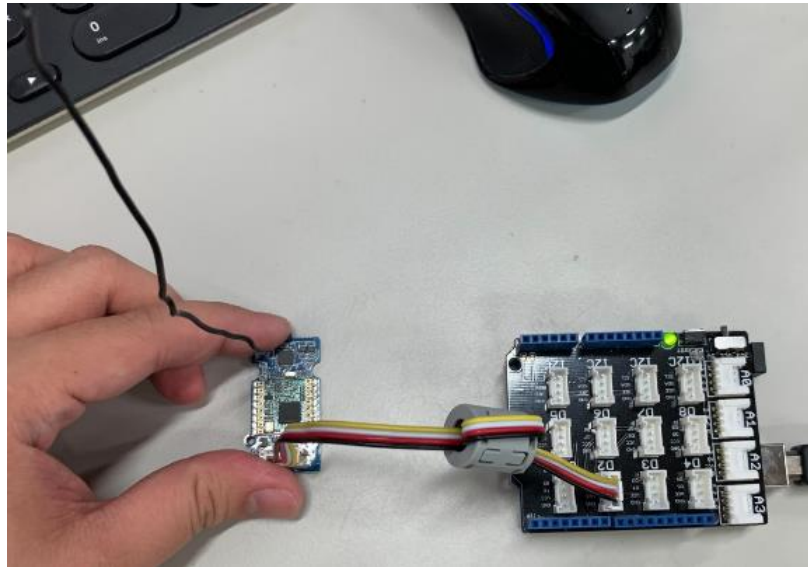


圖 5. Lora 模組接線圖。

(2) 打上以下程式碼。

```
////////// 主端程式//////// 負責接收訊息
#include <RH_RF95.h>
#include <SoftwareSerial.h>
SoftwareSerial SSerial(2,3); // RX, TX
#define COMSerial SSerial
#define ShowSerial Serial
```

```

RH_RF95<SoftwareSerial> rf95(COMSerial);

void setup() {
    ShowSerial.begin(115200);
    ShowSerial.println("RF95 server test.");
    if (!rf95.init()) {
        ShowSerial.println("init failed");
        while (1);
    }
    rf95.setFrequency(433.0);
}

void loop() {
    if (rf95.available()) {
        // Should be a message for us now
        uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
        uint8_t len = sizeof(buf);
        if (rf95.recv(buf, &len)) {

            ShowSerial.print("got request: ");
            ShowSerial.println((char*)buf);

            // Send a reply
            /* uint8_t data[] = "And hello back to you";
            rf95.send(data, sizeof(data));
            rf95.waitPacketSent();
            ShowSerial.println("Sent a reply");*/

        } else {
            ShowSerial.println("recv failed");
        }
    }
}

```

(3) 編譯成功後即可從序列埠看到運行狀態，如圖 6 所示。

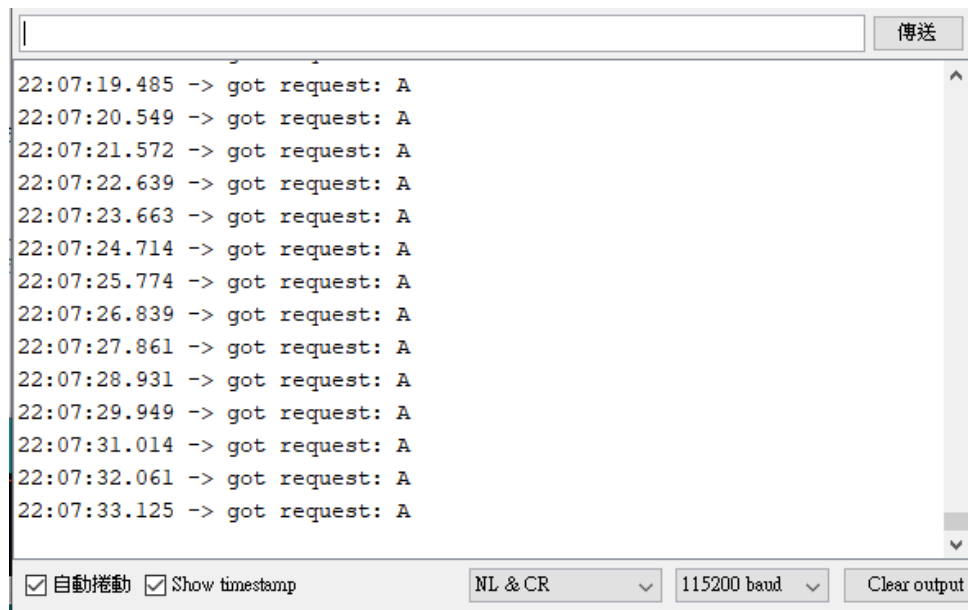


圖 6.收到主端發送的訊息。

### †實驗 3- 指定主從端訊息傳送程式步驟

1. 將從端程式燒錄至 Arduino。

(1) 如下圖連接 LoRa 模組(VCC、GND、TX(2)、RX(3))。

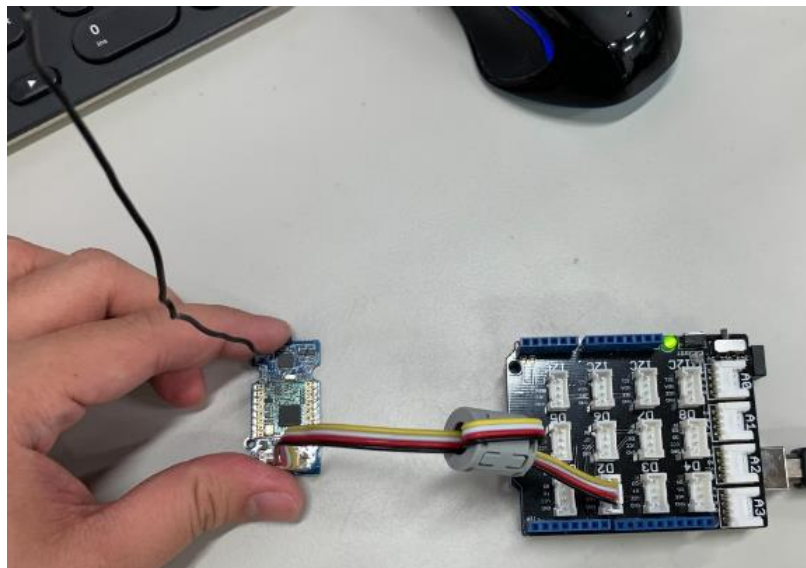


圖 7. Lora 模組接線圖。

(2) 請以兩組

(3) 第二組為 B(等於 ASCII 碼的 66)，後續訊息請各組自行設定，以此列推。

Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr
0	0	000	NUL (null)	32	20	040	Space	64	40	100	@	96	60	140	`
1	1	001	SOH (start of heading)	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX (start of text)	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX (end of text)	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT (end of transmission)	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ (enquiry)	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK (acknowledge)	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL (bell)	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS (backspace)	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB (horizontal tab)	41	29	051	)	73	49	111	I	105	69	151	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	B	013	VT (vertical tab)	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	D	015	CR (carriage return)	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	E	016	SO (shift out)	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	F	017	SI (shift in)	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE (data link escape)	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1 (device control 1)	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2 (device control 2)	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3 (device control 3)	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4 (device control 4)	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN (synchronous idle)	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB (end of trans. block)	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN (cancel)	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM (end of medium)	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB (substitute)	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC (escape)	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	FS (file separator)	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS (group separator)	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	RS (record separator)	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US (unit separator)	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

圖 8. ASCII 碼表。

程式碼：

```

////////// 主端程式
#include <RH_RF95.h>
#include <SoftwareSerial.h>

int X;//辨識字節

SoftwareSerial SSerial(2,3); // RX, TX

#define COMSerial SSerial
#define ShowSerial Serial

RH_RF95<SoftwareSerial> rf95(COMSerial);

void setup() {
  ShowSerial.begin(115200);
  ShowSerial.println("RF95 server test.");
  if (!rf95.init()) {
    ShowSerial.println("init failed");
    while (1);
  }
  rf95.setFrequency(433.0);
}

void loop() {
  if (rf95.available()) {

```

```

// Should be a message for us now
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);
if (rf95.recv(buf, &len)) {
    X=(buf[0]);///辨識第一個字節
    if (X==65){
        ShowSerial.print("got request: ");
        ShowSerial.println((char*)buf);
    }
    // Send a reply
    /* uint8_t data[] = "And hello back to you";
    rf95.send(data, sizeof(data));
    rf95.waitPacketSent();
    ShowSerial.println("Sent a reply");
*/

    } else {
        ShowSerial.println("recv failed");
    }
}

}

//////////從端程式
#include <RH_RF95.h>
#include <SoftwareSerial.h>
SoftwareSerial SSerial(2, 3); // RX, TX
#define COMSerial SSerial
#define ShowSerial Serial
RH_RF95<SoftwareSerial> rf95(COMSerial);

void setup() {
    ShowSerial.begin(115200);
    ShowSerial.println("RF95 client test.");

    if (!rf95.init()) {
        ShowSerial.println("init failed");
        while (1);
    }
    rf95.setFrequency(433.0);
}

void loop() {

```



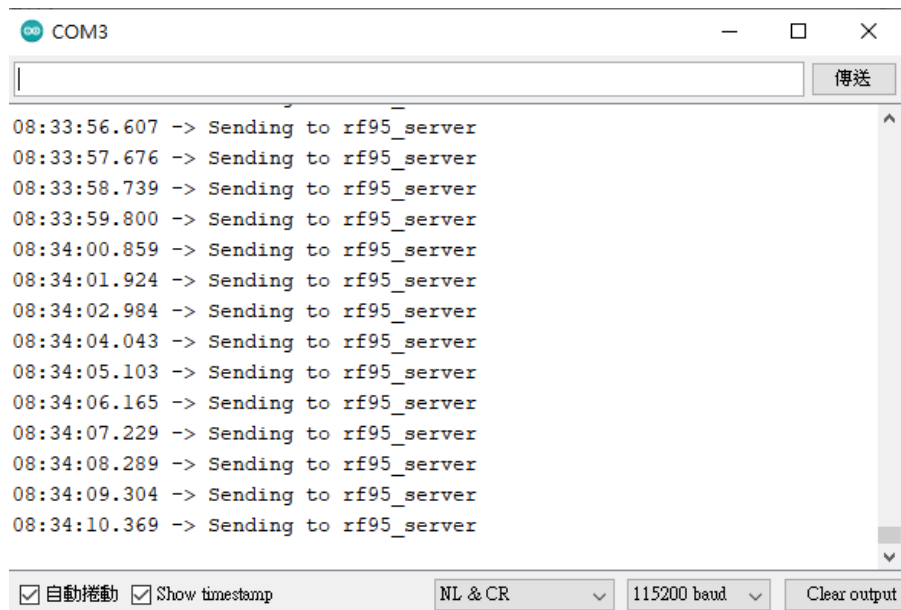
```

ShowSerial.println("Sending to rf95_server");
// Send a message to rf95_server

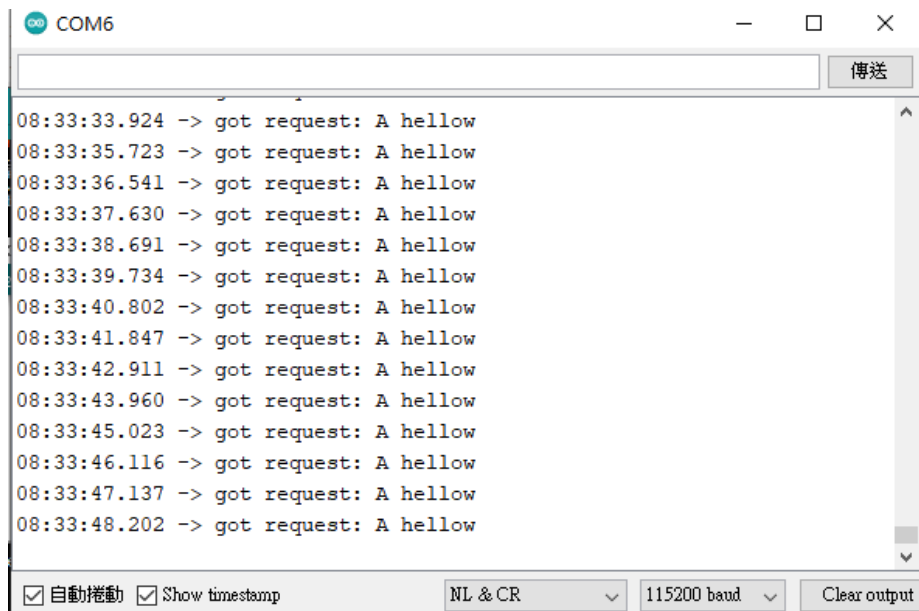
/*if (ShowSerial.available()) {
    val = ShowSerial.read();
}
ShowSerial.print(val);*/
uint8_t data[] = "A hellow";///設定訊息
rf95.send(data,sizeof(data));
rf95.waitPacketSent();
// Now wait for a reply
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);
/*
if (rf95.waitForAvailable(3000)) {
    // Should be a reply message for us now
    if (rf95.recv(buf, &len)) {
        ShowSerial.print("got reply: ");
        ShowSerial.println((char*)buf);
    } else {
        ShowSerial.println("recv failed");
    }
} else {
    ShowSerial.println("No reply, is rf95_server running?");
}
*/
delay(1000);
}

```

(4) 編譯成功後即可從序列埠看到運行狀態，如圖 9 所示。



(a)

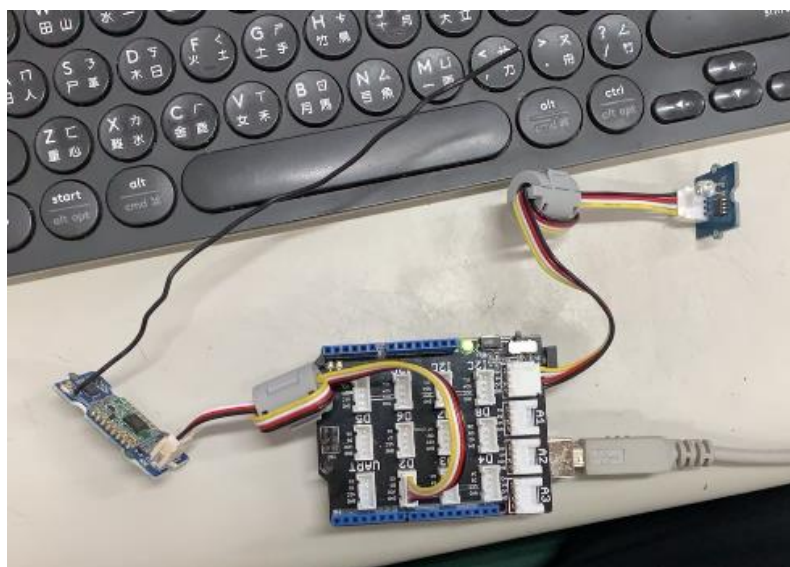


(b)

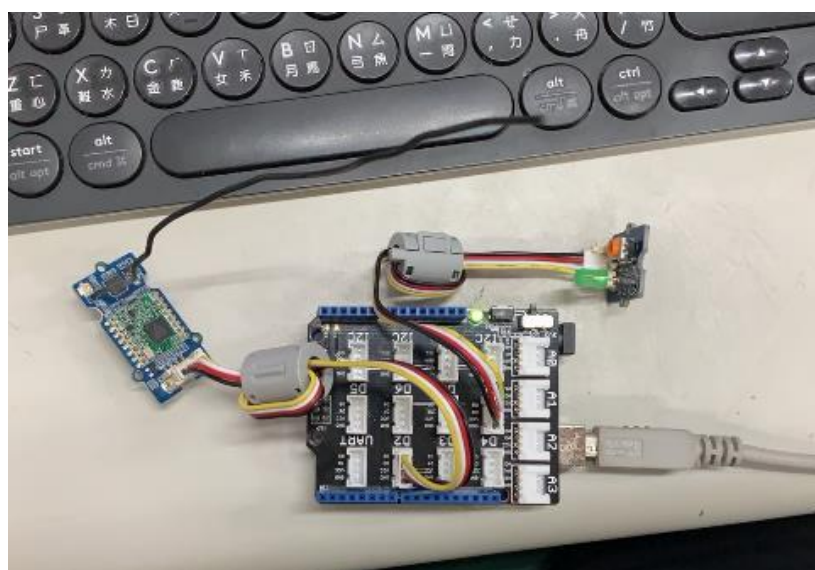
圖 9.(a) 從端發送訊息狀態。(b)主端接收訊息。

◆自己動手做 1：請修改上面實驗 3 的程式碼，請主端收到訊息後，進行回傳訊息給指定從端，而從端要顯示其訊息。

◆自己動手做 2：請修改上面實驗 3 的程式碼，請從端方接上亮度感測器，如圖 10(A)所示，主端接上一顆 LED 燈在有 PWM 功能的腳位上如圖 10(B)所示，而從端方會將讀取亮度類比值傳給指定主端，而主端方並根據從端方的亮度顯示越低同時顯示在序號列上，主端方的 LED 燈要越亮，反之相反，並回傳其目前給 LED 燈的 PWM 值給從端方。



(a)



(b)

圖 10.(a) 從端接線圖。(b)主端接線圖。

◆作業 2：回答問題

1. 解釋 `rf95.recv()` 及 `rf95.send()` 的用法及特性。
2. 請上網查詢 LoRa 普遍的封包格式，和其規格做說明。
3. 請說明這次實驗中的 LoRa 封包格式為何？